Renewable and Sustainable Energy Reviews 27 (2013) 20–29

Contents lists available at SciVerse ScienceDirect

Renewable and Sustainable Energy Reviews

journal homepage: www.elsevier.com/locate/rser

A comparative study of optimal hybrid methods for wind power prediction in wind farm of Alberta, Canada

Nooshin Bigdeli*, Karim Afshar, Amin Shokri Gazafroudi, Mostafa Yousefi Ramandi

EE Department, Imam Khomeini International University, Daneshgah Blv, Qazvin, Iran

A R T I C L E I N F O

Article history:
Received 5 February 2013
Received in revised form
2 June 2013
Accepted 16 June 2013
Available online 16 July 2013

Keywords:
Genetic algorithm (GA)
Imperialist competitive algorithm (ICA)
Neural networks
Particle swarm optimization (PSO)
Recurrence plots
Time series analysis

A B S T R A C T

In the recent years, by rapid growth of wind power generation in addition to its high penetration in power systems, the wind power prediction has been known as an important research issue. Wind power has a complicated dynamic for modeling and prediction. In this paper, different hybrid prediction models based on neural networks trained by various optimization approaches are examined to forecast the wind power time series from Alberta, Canada. At first, time series analysis is performed based on recurrence plots and correlation analysis to select the proper input sets for the forecasting models. Next, a comparative study is carried out among neural networks trained by imperialist competitive algorithm (ICA), genetic algorithm (GA), and particle swarm optimization approach. The simulation results are representative of the out-performance of ICA in tuning the neural network for wind power forecasting.

Contents

* Corresponding author. Tel.: +98 281 8371759/+98 912 1820372; fax: +98 281 8371761/+98 281 837 1155.
E-mail addresses: bigdeli@ikiu.ac.ir, nooshin_bigdeli@yahoo.com (N. Bigdeli), afshar@ikiu.ac.ir (K. Afshar), shokri_amin@ymail.com (A.S. Gazafroudi), mostafa_yousefi64@yahoo.com (M.Y. Ramandi).

1364-0321/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.rser.2013.06.022

1. Introduction

Electrical power generation is increased caused by population growth and its subsequent aggressive electrical energy demands [1]. Thermal pollution is increased and greenhouse gases are produced more, due to the growth of electrical energy generation resulting from thermal power plants. It causes more interest in power generation based on renewable energies [2]. Electrical power generation based on wind energy has been fastest growing among the renewable energy sources [3]. It is estimated that in 2020, about 12% of the world electrical energy will be supplied from wind energy [4]. Therefore, the electricity generated by wind power will play an important role in electricity supply.

Wind power depends on weather conditions such as wind speed, wind direction, temperature, air pressure and environmental obstacles. As a dynamic system, wind power has a correlation with its past values at any time, as well [3]. Due to the dependence of wind power on the atmospheric parameters, it has been recognized as a non-dispatchable source [5]. This feature introduces a wind power as an uncertain variable and reduces the system reliability. Therefore, an accurate prediction of wind power variations can moderate this problem to some extent [6,7].

Wind power prediction based on meteorological variables is encountered with some difficulties. That is, sufficiently accurate measurements of meteorological variables are commonly unavailable and their measurement equipments are so expensive to be supported, elsewhere. Inaccurate measurements or estimations can, on the other hand, results in aggressive errors in the wind power forecasting. As another fact, the true model of the wind power generating unit is not in hand, commonly. Therefore, achieving a low wind power forecasting error via a relatively simple black-box model with a low number of measurable inputs/input variables is perfectly desired.

Based on the above discussion, in this paper, wind power forecasting based on its historical data as the forecasting model inputs is considered. That is, the optimal training of neural networks is proposed as our modeling approach and four seasonal wind power data sets of Alberta, Canada [8] wind farm are studied as the real data for model construction and evaluation. In order to construct the neural network model for forecasting of the wind power, at first, time series analysis is performed based on recurrence plots and correlation analysis to the available wind power time series. In the next stage, a comparative study is carried out among various neural networks trained by imperialist competitive algorithm (ICA) [9], genetic algorithm (GA) [10], and particle swarm optimization (PSO) [11,12] approach. The simulation results are representative of out-performance of ICA in tuning the neural network for wind power forecasting.

This paper is organized as follows. In Section 2, the related researches are introduced. In Section 3, the data properties and the input selection approach is described. In Section 4, the proposed wind power prediction engine is presented. In Section 5, design and evaluation of the forecasting models for the wind power time series of Alberta, Canada are described. Finally, Section 5 concludes the paper.

2. The related researches

Wind power forecasting methods can be categorized as the physical and time series or statistical models [13,14]. In the physical modeling, someone tries to estimate the wind speed time series taking into account the physical characteristics of the environment conditions [15]. The statistical model is attempted to find a relationship between the parameters of the historical data to predict the future wind speed and wind power [16]. Commonly, physical models are used for long-term prediction and statistical model are used for short-term prediction [17].

In the literature, there are different attempts for short-term wind power forecasting via hybrid time series methods. In [18], wind power prediction has been done via a composition of modified hybrid neural network and enhanced particle swarm optimization algorithm. In [19], wavelet transform support vector machine in conjunction with statistic-characteristics analysis has been employed for short-term wind power prediction. In [20], a method has been presented to improve the short-term wind power prediction at a given turbine using information from numerical weather prediction and from multiple observation points. In this paper, the prediction of wind power is achieved in two stages; in the first stage wind speed is predicted using the proposed method. In the second stage, the wind speed to output power conversion is accomplished using power curve model. In [21], a useful model based on wavelet transform, chaotic time series and the GM (1,1) method has been presented for wind farm power forecasting. A new approach based on clustering has been proposed in [22] and in [23], the ultra-short term prediction of wind power based on chaotic time series has been considered. Artificial neural networks (ANN) optimized by Tabu search algorithm [24], hybrid PSO-ANFIS approach [25], wind farm power generation based on fuzzy modeling [26], and a hybrid strategy of short term wind power prediction based on the physical strategy and ANN technique [27] have been addressed in the literature as well. Besides, comprehensive reviews about the methods and models of wind power may be found in [28–30].

3. The data properties and selection of appropriate input set

As stated earlier, in this paper, the prediction of wind power experimental data from Alberta, Canada wind farm [8] is considered. The available data are four seasonal data sets for year 2007, each one containing 1368 hourly stored data. The wind power is predicted using feed-forward neural networks trained by some optimization algorithms being ICA, GA and PSO. In the feed-forward neural networks, the outputs at any moment only depend on the neural weights and the input signals to the neural network at that moment. Therefore, proper selection of inputs is essential to obtain good performance of the trained neural network. To do that, in this paper, two stages are followed to determine the neural network inputs for each seasonal data set. At the first stage, the characteristics and predictability of the wind power time series is investigated via recurrence plots. Based on the derived results, in the next stage, the correlation analysis is performed to choose proper input sets for the four seasonal data sets.

3.1. The available data and its properties

Seeking for the proper inputs for our models, in this section the experimental data from Alberta, Canada wind farm [8] for year 2007 will be examined, closely. As mentioned earlier, the available data are four seasonal data sets, each one containing 1368 hourly stored data. The mentioned data have been shown in Fig. 1(a)–(d). As shown in this figures, severe fluctuations is observed in the wind power time series while no hallmark of strong periodicity is demonstrated. However, such fluctuations may be due to the chaotic or stochastic nature of a nonlinear process [31–33]. Since, we are interested in predictability, it is important for us to distinguish between these two types of processes. This property has been closely examined by the authors in [34] via time series analysis methods, where the results are representative of stochastic nature and so short-term predictability of wind power time series in short-term time scale. In order for briefly representing

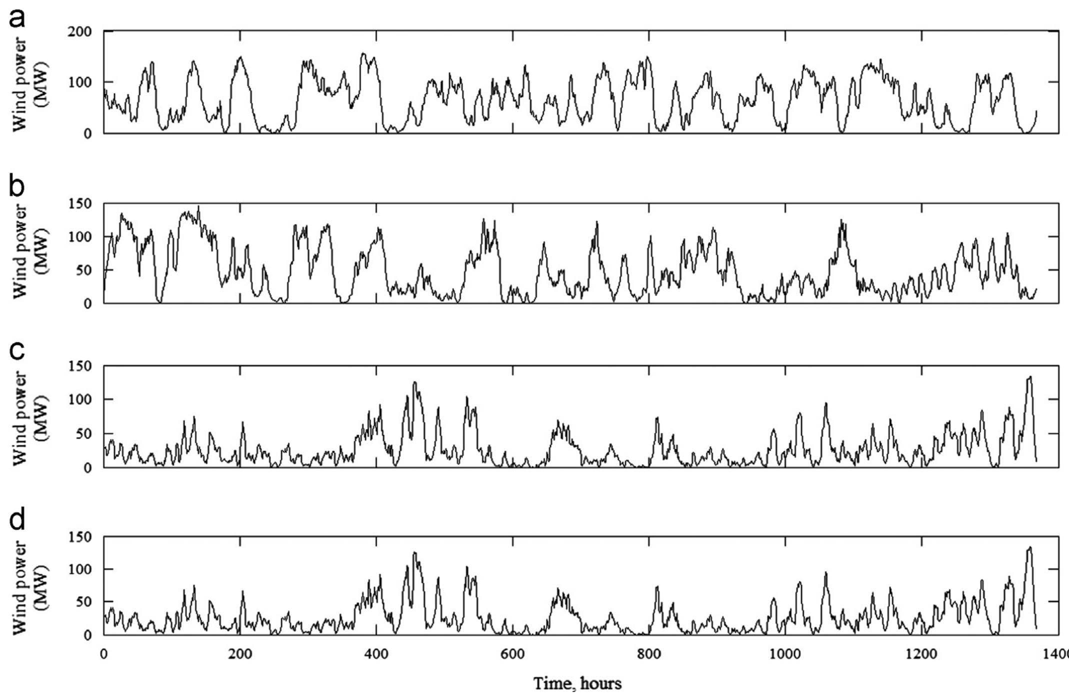


Fig. 1. The Alberta wind power time series for (a) the 1st season, (b) the 2nd season, (c) the 3rd season and (d) the 4th season.

the results in [34] and to get a better view about the behavior and characteristics of the underlying system, the Recurrence Plots (RPs) of wind power time series are investigated, in this section.

3.1.1. *The fundamentals of recurrence plots*

Recurrence is a fundamental property of dynamic systems, which can be exploited to characterize the system's behavior. As a powerful tool for the visualization and analysis of the phase space trajectory of the experimental time-series, recurrence plot (RP) was introduced in the late 1980s by Eckman et al. [35]. It is especially useful for finding hidden correlations in highly complicated data and to determine the stationarity of the time series [36]. With RP, one can graphically detect hidden patterns and structural changes in data or see similarities in patterns across the time series under study [37]. This technique has been successfully applied to various fields, such as physiology [38,39], fluid dynamics [40], geology [41], economy [42], as well as energy market indices [43–45]. In this paper, the RP methodology will be applied to analyze the wind speed time series behavior. Especially the predictability of the wind time series would be investigated via these analyses.

For deriving an RP, first of all the phase space of signal must be reconstructed via say "method of delays [46]". RPs visualize the behavior of trajectories in phase space [36,47,48] via a graphical representation of the matrix:

$$R_{i,j} = \Theta(\varepsilon - \| \vec{x}_i - \vec{x}_j \|) \quad i,j = 1, \ldots, N \qquad (1)$$

where, $\vec{x}_i$ stands for the point in the reconstructed phase space at time i, and ε is a predefined threshold and $\Theta(.)$ is the Heaviside function. One assigns a "black" dot to the value one and a "white" dot to the value zero. The two-dimensional graphical representation of $R_{i,j}$ then is called RP [47] and can be used to distinguish between different dynamic systems. In this context, recurrence plot (RP) examines the paths in the state space. Three types of systems are recognized based on the obtained curve: (1) Periodic systems, (2) Stochastic systems and (3) Chaotic systems [47,48]. Periodic systems are marked by parallel lines and non-interrupted diametric, where distance between the lines is proportional to the period. These diametric lines are also seen in chaotic systems, but the lines have been cut and their length is shorter. Also, the distance between these lines is irregular. The lengths of lines are proportion to the degree of system predictability. RP curves of uncorrelated stochastic systems consist of many individual dots that their distribution is quite irregular [47,48].

Table 1
The embedding delay and embedding dimension for the Alberta wind power time series.

Season #	1	2	3	4
Embedding delay	5	7	7	6
Embedding dimension	20	14	16	16

3.1.2. *The recurrence plot analysis results*

In order to reconstruct the phase space of the wind time series, initially, the embedding delay and the embedding dimension of the time series must be acquired. The mutual information method [46], and the false nearest neighbors method have been used to calculate the embedding delay and embedding dimension of the fours seasonal wind power time series. These embedding time delays and embedding dimensions are expressed at Table 1. The RP will be achieved by using the dimensions and delays of these time series.

The RPs of wind power time series are shown in Fig. 2(a)–(d) plotted via the CRP toolbox of MATLAB [49] as our tool. Concerning these figures it is concluded that for the first and second seasons, the short term erratic distribution of recurrence points is representative of strong stochastic nature of the underlying time series with mimic predictability. The situation is somehow different in seasons 3 and 4, where the recurrence diagonals are longer and thus, the predictability would be increased. White ribbons in the recurrence plots correspond to transitions in the system dynamics. Such dynamic transitions as well as various seasonal properties

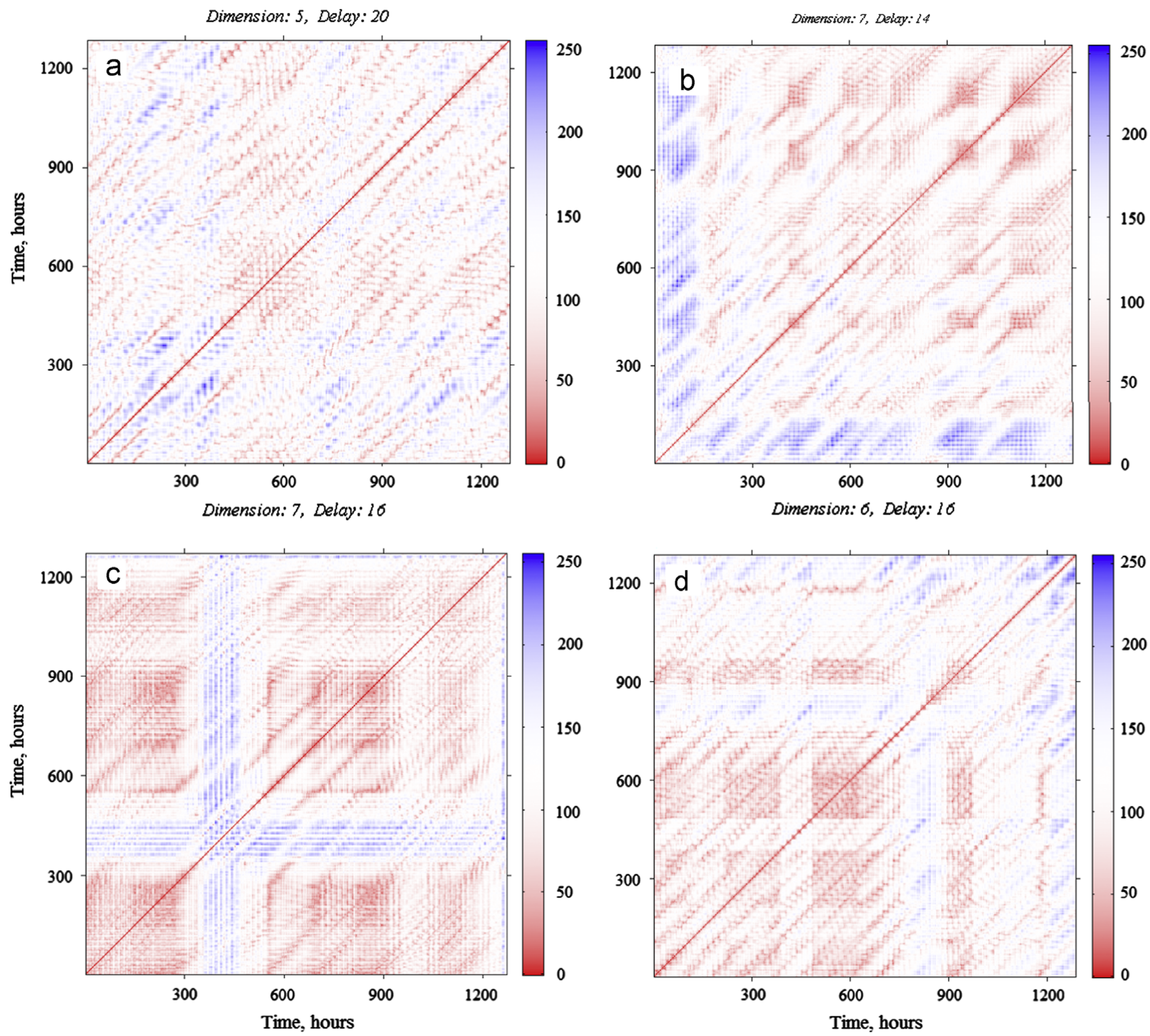


Fig. 2. The RP of the Alberta wind power time series for (a) the 1st season, (b) the 2nd season, (c) the 3rd season and (d) the 4th season.

are representative of the seasonality as well as non-stationarity of the wind power time series. Therefore, in selecting the inputs for the forecasting model, the mimic predictability of the wind time series should be taken into account. Besides, since the forecasting model inputs are the lagged wind power terms, they should be as close as possible to the desired time to compensate for the non-stationarity of the dynamics.

Based on the above discussions, in the following section, the correlation analysis is carried out to select the appreciate inputs for the forecasting model.

3.2. Correlation analysis

Once the mimic predictability of interested wind time series is concluded, we should analyze the correlation properties the available data to choose the proper model inputs. The plots in Fig. 3(a)–(d) show the autocorrelation function plot of the seasonal wind power data sets.

In these figures, it is illustrated that the wind power in each hour is highly correlated with its lagged values in the same day up to a few hours. For the previous days, the correlation decays aggressively, which is another hallmark of mimic predictability. We adopt a threshold of 0.7 of correlation to select the model inputs. This threshold corresponds to 6, 6, 4, and 6 lagged values for the four seasons, respectively.

4. The power prediction engine

Regarding the high performance of neural networks in modeling of nonlinear dynamics, in this paper, they have been employed as our modeling tool for wind power prediction. In this section, we shortly review the basics of neural networks and then switch to the developed models and their performance.

4.1. The fundamentals of ANN's

Neural networks are highly interconnected simple processing units designed in a way to model how the human brain performs particular task [50,51]. Each of those units, called neurons, forms a weighted sum of its inputs, to which a constant term called bias is added. This sum is then passed through a transfer function: linear, sigmoid or hyperbolic tangent (Fig. 4(a)).

In a typical ANN, the neurons are organized in a way that defines the network architecture. Networks with interconnections that do not form any loops are called feed-forward. Recurrent or

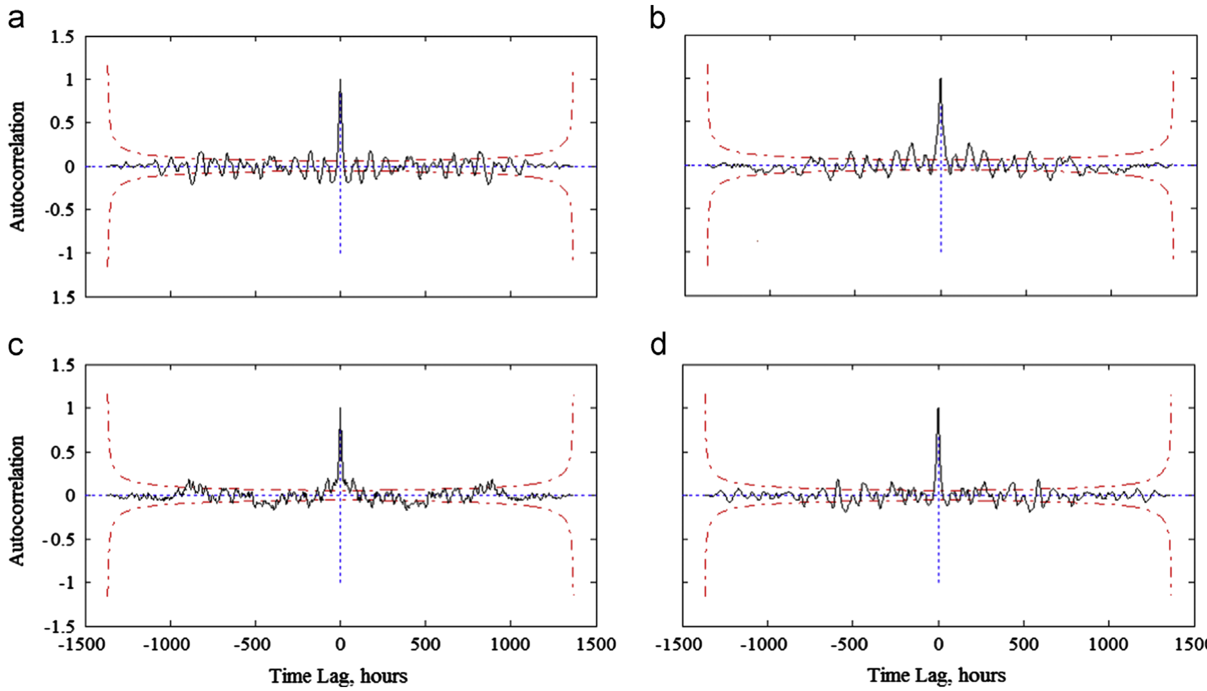


Fig. 3 SEQ Figure * ARABIC. The autocorrelation of Alberta wind power time series for (a) the 1st season, (b) the 2nd season, (c) the 3rd season and (d) the 4th season.

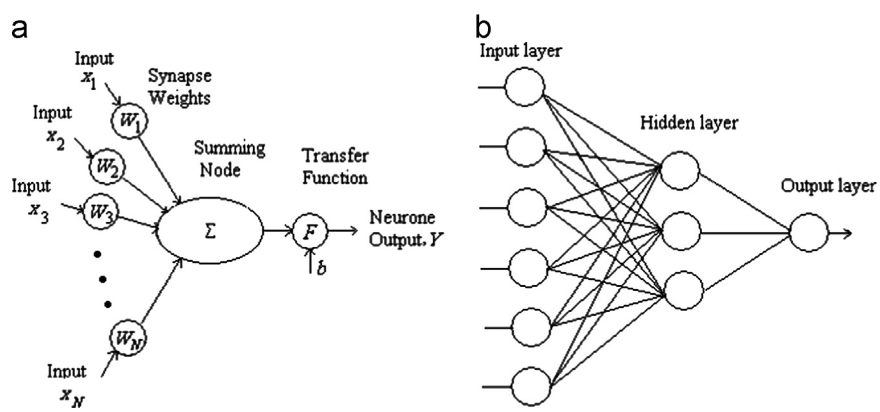


Fig. 4. (a) Internal structure of a neuron and (b) the structure of an example three layer feed-forward neural network.

non-feed forward networks in which there are one or more loops of interconnections are also used for some kinds of applications [52]. Multilayer perceptrons (MLP's) are the best known and most widely used kind of ANNs. In these neural networks, neurons are arranged in layers: an input layer, one or more hidden layers and an output layer. The neurons in each layer may share the same inputs, but are not connected to each other. Fig. 4(b) shows the architecture of a generic three-layered feed-forward neural network model. In order to find the optimal network architecture, one should evaluate several combinations. These combinations include in networks with different number of hidden layers, different number of neurons in each layer and different types of transfer functions. Typically, the number of neurons in the hidden layer is chosen by trial and error.

In the feed forward neural networks, the output only depends on input signals and neural weights at that moment. The activation function used in the hidden layers is commonly nonlinear transfer functions such as and log-sigmoid function with its output in [0, 1] interval, or tan-sigmoid function for penning the input to the interval $[-1, 1]$. The output of a hidden layer is compute as:

$$n_i = w_{i1}x_1 + w_{i2}x_2 + \ldots w_{iR}x_R + b_i \tag{2}$$

$$a_i = f(n_i), \quad i = 1, 2, 3, \ldots S \tag{3}$$

where, x_R is the Rth input; S is the number of neurons; w_{iR} is the related weight of the input vector and ith neuron of the hidden layer; b_i is its bias; and $f(.)$ is the activation function. The output is computed in the output layer in the same manner as the hidden layers unless the linear transfer function is commonly used in this layer as the activation function.

Forecasting with neural networks involves two steps: training and testing. Training of feed forward networks is normally performed in a supervised manner. In supervised manner both input and outputs are participated in training the network. The adequate selection of inputs for neural network training is highly influential to the success of training. A learning process in the neural network then constructs an input–output mapping by adjustment of the weights and biases at each iteration based on

the minimization of some error measure between the produced and the desired output. Thus, learning entails an optimization process. The knowledge acquired by the neural network through the learning process is tested by applying new data that it has never seen before, called the testing set. The network should be able to generalize and have an accurate output for this unseen data [43].

The most common learning algorithm is the back propagation algorithm [53], in which the error is propagated back to the input in order for adjusting the weights and biases in each layer. The standard back propagation learning algorithm is a steepest descent algorithm that minimizes the sum of square errors. This standard back propagation learning algorithm is not efficient numerically and tends to converge slowly [50,53]. An algorithm that trains a neural network 10–100 times faster than the usual back propagation algorithm is the Levenberg–Marquardt algorithm. The Levenberg–Marquardt algorithm is a variation of Newton's method [50]. Newton's update for minimizing a function $V(\vec{x})$ with respect to the input vector $\vec{x}$, is given by:

$$V(\vec{x}) = \frac{1}{N}\sum_{h=1}^{N} e_h^2(\vec{x}) \tag{4}$$

where, $e(\vec{x})$ is the output error vector. The details about the Levenberg–Marquardt algorithm can be found in [43]. This method, however, commonly suffers from lack of convergence to the global optimum. Therefore, employing a more efficient optimization algorithm may lead to more accurate response, less forecasting error as well as better convergence. Based on the above discussions, in the following sections, some optimizations algorithms being imperialist competitive algorithm (ICA), genetic algorithm (GA), and particle swarm optimization (PSO) approach are employed for training the neural network for forecasting the wind power time series from Alberta, Canada and the results are compared for this case.

4.2. The trainer unit

In this section the optimization algorithms employed for training the forecasting neural network models are briefly introduced.

4.2.1. Imperialist competitive algorithm

Imperialist competitive algorithm (ICA) is a new optimization technique that is inspired by imperialism countries competing social and political processes. ICA has shown its outstanding ability for the various problems [54–57]. This algorithm is initially started with N Clooney in which, N_{imp} is the best one (country with the lowest cost) which is selected as imperialisms. In [58], ICA pseudo-code is described as follows:

i. Selection of the random locations of the function and initialize the empires.
ii. Moving the colonies toward their related imperialist (absorption policy or assimilation) according to predetermined assimilation coefficient ($\beta > 1$) and assimilation angle coefficient (γ), which determine the angle and amount of movement.
iii. Changing randomly the location of colonies (revolution).
iv. Until the cost of colony is less than the imperialist, it remains in the empire and changes its location relative to imperialist.
v. Uniting the empires with the same conditions.
vi. Calculating the total cost of all empires via:

$$\text{Totalcostofempire} = \text{Costofimperialist} + \zeta \times \text{mean(costofallcolonies)} \tag{5}$$

where, ζ is a constant and mean(.) stands for the average of its arguments.
vii. Selecting the weakest colony (colonies) from the weakest empires and put it (them) in one of the empires (colonial competition).
viii. Destroying the weak empires.
ix. If the preset conditions satisfied, it will stop, otherwise return to 2.

4.2.2. Genetic algorithm

A genetic algorithm emulates biological evolution to solve optimization problems. It is formed by a set of individual elements (the population) and a set of biological inspired operators that can change these individuals. According to evolutionary theory only the individuals that are the more suited in the population are likely to survive and to generate off-springs, thus transmitting their biological heredity to new generations.

In computing terms, genetic algorithms map strings of numbers to each potential solution. Each solution becomes an individual in the population, and each string becomes a representation of an individual. There should be a way to derive each individual from its string representation. The genetic algorithm then manipulates the most promising strings in its search for an improved solution. The algorithm operates through a simple cycle [10]:

i. Creation of a population of strings.
ii. Evaluation of each string.
iii. Selection of the best strings.
iv. Genetic manipulation to create a new population of strings.

4.2.3. Particle swarm optimization algorithm

Particle swarm optimization (PSO) is a method for performing numerical optimization without explicit knowledge of the gradient of the problem to be optimized. PSO is originally attributed to Kennedy, and Eberhart was first intended for simulating social behavior [12]. The algorithm was simplified and it was observed to be performing optimization. PSO is an efficient population based optimization technique, which is appropriate for non-convex optimization problems [11,12]. In general, the velocity update of the ith particle at the k+1th iteration is expressed as [11]:

$$v_i^{k+1} = w \times v_i^k + c_1 \times r_1 \times (p_i^k - x_i^k) + c_2 \times r_2 \times (p_g - x_i^k) \tag{6}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{7}$$

where, in Eq. (6), v_i^k is the velocity of the ith particle at the kth itertaion,p_g is the swarm's best known position, w is the inertia weight, c_1, c_2 are the learning factors, and x_i^k is the position of the ith particle at the kth iteration. In Eq. (6), the first term provides the necessary momentum for particles to roam across the problem space. The second is the cognitive component that represents the individual experience of each particle. The second component encourages the particles to move toward their own best positions reached. The last component is the social collaboration of the particles in finding the global optimal solution. The particles are pulled toward the global best particle reached. Finally, the position of the ith particle is updated by Eq. (7) [11].

4.3. Evaluation indices

As stated earlier, for the evaluation of the ANN's performance, a testing set containing new input data that it has never seen before is applied to the trained network. The performance of the trained network is then evaluated by comparison of the network output with its actual value. There are some statistical evaluation indices which are commonly used to judge about an ANN's performance. Let A_i and P_i be the actual and network output, respectively,

related to ith input vector, where N is the number of points in the testing set. Then the evaluation indices are defined as [43]:

- Mean absolute error (MAE):

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|P_i - A_i| \tag{8}$$

- Root mean square error ($RMSE$):

$$RMSE = \frac{1}{N}\sqrt{\sum_{i=1}^{N}(P_i - A_i)^2} \tag{9}$$

- Mean absolute percentage error ($MAPE$):

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|P_i - A_i|}{A_i} \times 100\% \tag{10}$$

- Modified mean absolute percentage error (Modified_MAPE): In the relationship in Eq. (10) if the actual value is large and its prediction becomes small, the computed relative error will become near 100%. On the other hand if the actual value is small, the relative error may become very large even though the difference is small. In this case, the relationship in Eq. (10) is modified in this manner. At first, the average of actual output values is computed as:

$$Av = \frac{1}{N}\sum_{i=1}^{N}A_i$$

and then, the Modified_MAPE will be computed as [59]:

$$Modified_MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|P_i - A_i|}{Av} \times 100\% \tag{11}$$

- Modified peak absolute percentage error ($Modified_PAPE$):

$$Modified_PAPE = Max\left(\frac{|P_i - A_i|}{Av}\right) \times 100\% \quad \text{for } 1 \le i \le N \tag{12}$$

5. Design and evaluation of the forecasting models

5.1. Input selection

The multi layer perceptron feed forward neural network with two hidden layers is proposed in this paper for short-term forecasting of the wind power time series. Based on the performed analyses and considering the short-term predictability of the wind time series as well as its non-stationarity and seasonality, four separate neural network models has been synthesized in order for forecasting the Alberta wind time series in each season. In order to select the appreciate inputs, a threshold of 0.7 has been considered to determine the correlated lagged data as the network inputs. The lagged data and the underlying correlations have been presented in Tables 2–5, corresponding to the auto-correlation graphs shown in Fig. 3. In these tables, WP_i, $i=1,..,4$, stands for the wind power time series of season i. Based on these results, the wind power of 1 to 6 hours before the desired hour has been considered as the neural network models' inputs for the first, second and fourth seasons, while it drops to the 1–4 h ago for the third season.

In order to find the optimal network input set, several correlation thresholds were evaluated. Amongst, the selected threshold and so the selected input set, in one hand, considers the correlation properties of the available data and, on the other hand, implies a proper convergence rate.

Table 2
Selected inputs, $WP_1(t)$, for the Alberta wind power time series, 1st Season.

Rank	Selected inputs	Auto-correlation	Rank	Selected inputs	Auto-correlation
1	$WP_1(t-1)$	0.976	4	$WP_1(t-4)$	0.826
2	$WP_1(t-2)$	0.933	5	$WP_1(t-5)$	0.766
3	$WP_1(t-3)$	0.881	6	$WP_1(t-6)$	0.706

Table 3
Selected inputs, $WP_2(t)$, for the Alberta wind power time series, 2nd Season.

Rank	Selected inputs	Auto-correlation	Rank	Selected inputs	Auto-correlation
1	$WP_2(t-1)$	0.973	4	$WP_2(t-4)$	0.831
2	$WP_2(t-2)$	0.928	5	$WP_2(t-5)$	0.781
3	$WP_2(t-3)$	0.881	6	$WP_2(t-6)$	0.733

Table 4
Selected inputs, $WP_3(t)$, for the Alberta wind power time series, 3rd season.

Rank	Selected inputs	Auto-correlation	Rank	Selected inputs	Auto-correlation
1	$WP_3(t-1)$	0.956	3	$WP_3(t-3)$	0.797
2	$WP_3(t-2)$	0.88	4	$WP_3(t-4)$	0.712

Table 5
Selected inputs, $WP_4(t)$, for the Alberta wind power time series, 4th season.

Rank	Selected inputs	Auto-correlation	Rank	Selected inputs	Auto-correlation
1	$WP_4(t-1)$	0.973	4	$WP_4(t-4)$	0.831
2	$WP_4(t-2)$	0.929	5	$WP_4(t-5)$	0.779
3	$WP_4(t-3)$	0.881	6	$WP_4(t-6)$	0.726

5.2. Network configuration

As stated earlier, in training an ANN, the number of hidden layers, and the number of the neurons of each layer affect the prediction precision and training rate, considerably. Therefore, in order to find the optimal network architecture, several combinations of inputs were evaluated. These combinations included networks with different number of hidden layers, different number of neurons in each layer and different types of transfer functions. We converged to a configuration consisting of two hidden layers and number of neurons as: 6 for input layer for the first, second and fourth seasons, and 4 for the third season, 7 and 5 for hidden layers and 1 for output layer. All of the input data were normalized between −1 and 1. Based on this normalization, the transfer function for input and hidden layer neurons has been selected as a tan-sigmoid transfer function, defined by:

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \tag{11}$$

The linear transfer function is also used in the neurons of output layer. For training the network, the neural network toolbox of MATLAB [60] was selected due to its flexibility and simplicity [9]. The cost function of Eq. (4) is considered as the training index, and ICA, GA, and PSO have been employed to find the optimal network weights to minimize the cost function. The corresponding properties and parameters of the optimization approaches have

been brought in Table 5. In order for training of each seasonal network, 1200 data of each wind power data set have been considered for training and 168 data have been used for evaluation.

5.3. Evaluation results

In this section, the performance of the proposed prediction engine is investigated. That is the performance of the neural network trained by ICA, PSO and GA are compared for wind power prediction. Figs. 5–8 show the results of the trained neural networks for the three cases. For comparison, the results for the method in [8] has been brought, as well. Besides, in Table 6 the validation indices i.e. *MAE*, *RMSE*, *Modified_MAPE* and *Modified_PAPE* for both test and train data have been brought. As seen from these results, the proposed prediction engine performs superior with respect to the method in [8]. Among the three proposed

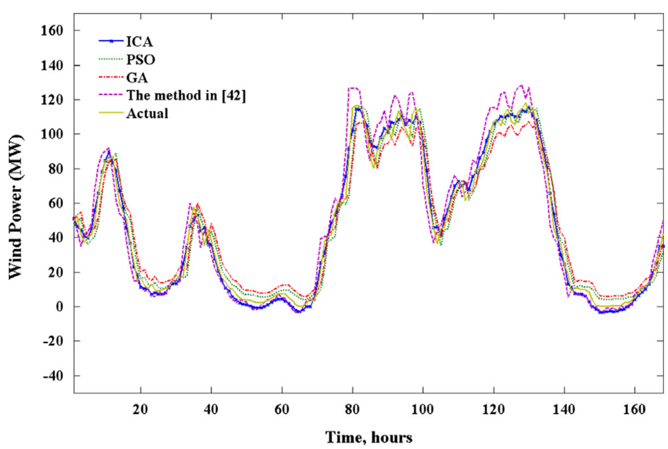


Fig. 5. The actual and forecasted wind power time series forecasted by hybrid *NN* for 1st test week of Alberta.

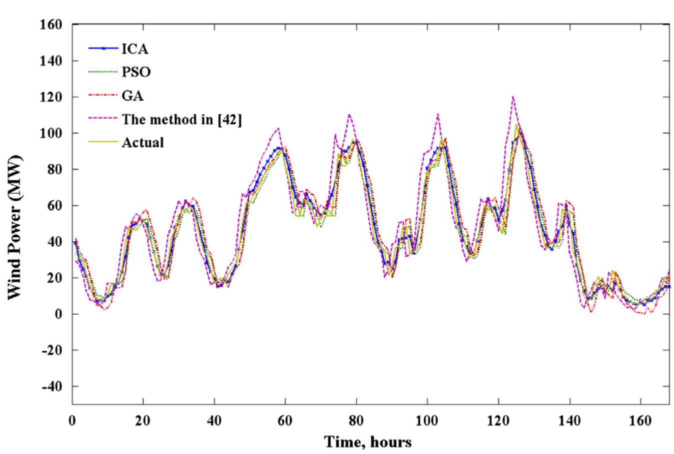


Fig. 6. The actual and forecasted wind power time series forecasted by hybrid *NN* for 2nd test week of Alberta.

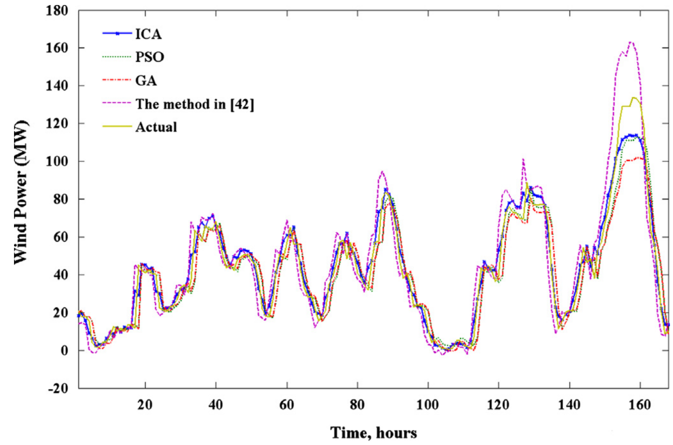


Fig. 7. The actual and forecasted wind power time series forecasted by hybrid *NN* for 3rd test week of Alberta.

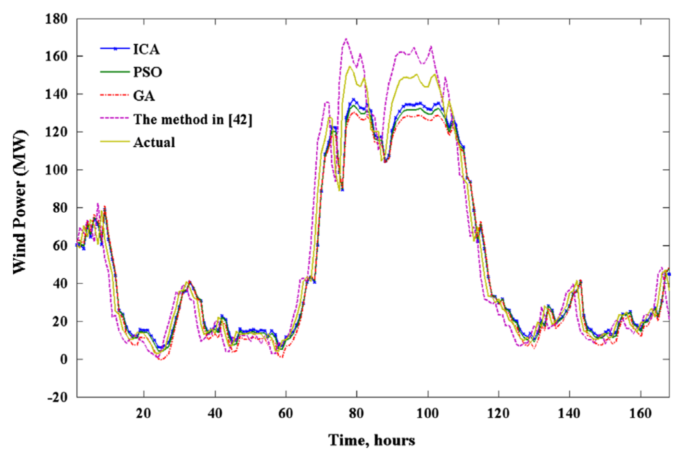


Fig. 8. The actual and forecasted wind power time series forecasted by hybrid *NN* for 4th test week of Alberta.

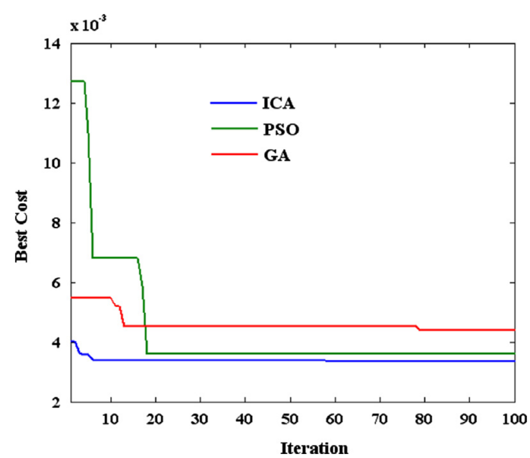


Fig. 9. The convergence curve of the proposed hybrid neural networks for various optimization methods.

Table 6
The properties and parameters of the employed optimization algorithms.

ICA		PSO		GA	
Number of initial countries.	40	Population size (swarm size)	200	Population size	100
Number of initial imperialists.	8	Personal learning coefficient (c_1)	2	Crossover percentage	0.7
Revolution rate	0.3	Global learning coefficient (c_2)	2	Mutation percentage	0.2
Assimilation coefficient (β)	2	Inertia weight damping ratio (w)	0.99		
Assimilation angle coefficient (γ)	0.5				
ζ	0.02				

Table 7
The comparison of the evaluation indices for different hybrid methods.

Method	Test weeks	Correlation of test data	*MAE* of test data	*RMSE* of test data	*Modified_MAPE* of test data	*Modified_PAPE* of test data
ICA-NN	1	0.99493	3.4320	4.2963	7.3888	27.6514
	2	0.98765	3.5459	4.5374	7.8355	28.9272
	3	0.97657	4.7084	6.6030	10.9268	47.3932
	4	0.98407	6.9516	9.5403	13.2342	87.6471
PSO-NN	1	0.98136	5.8490	7.6116	12.5770	59.3311
	2	0.95684	5.9023	7.5083	13.0523	43.7229
	3	0.94692	6.9452	9.7647	16.1152	77.0529
	4	0.98296	7.2965	10.1025	13.9086	87.1824
GA-NN	1	0.98146	7.0851	8.5152	15.2348	65.3294
	2	0.95513	7.0337	8.5009	15.5544	53.9946
	3	0.93777	7.4237	10.8097	17.2253	78.37353
	4	0.98040	8.2172	11.1027	15.6636	85.7794
The method in [8]	1	0.9791	6.9984	9.2971	15.0669	87.3696
	2	0.9319	7.2740	9.6440	16.0733	66.6019
	3	0.90547	8.7586	12.3679	20.3263	108.9452
	4	0.9624	7.7107	13.8326	15.0669	109.7564

hybrid cases, the hybrid of ICA and *NN* shows the best performance with the lowest error indices. From convergence point of view, the methods have been compared in Fig. 9. That is, ICA in conjunction with neural network show the fastest convergence, while the neural network model trained by PSO is faster than the model trained by GA Table 7.

6. Conclusions

In this paper, accurate forecasting of wind power, as a key requirement to acquire proper performance of a wind farm has been considered. The desired wind power to forecast are the four seasonal wind power data sets of Alberta, Canada wind farm which are studied as the real data for model construction and evaluation. In order to synthesize an accurate model for wind power prediction, at first, the wind power time series behavior has been characterized via a powerful time series analysis method known as recurrence plot. Via this characterization, it is observed that the wind time series exhibit as stochastic signal with mimic predictability. The non-stationarity and seasonality of this time series are the other characteristics of the wind power. Based on the analysis results short-term forecasting of the wind time series has been considered via some hybrid optimized neural network models. Due to the mimic predictability of the time series the close past values of the time series which are highly correlated with the hourly wind power time series have been considered as the model inputs. Such correlation analyses has lead to selection of the wind power at most 1–6 h before the desired as the neural network models' inputs. Next, the neural network model has been trained via three powerful optimization algorithms which are GA, PSO and ICA. The prediction results as well as the evaluation indices are representative of the out-performance of the hybrid model of neural network and ICA with respect to others. Low error indices and very fast convergence are the main properties of the hybrid ICA-neural network model.

References

[1] Johnson PL, Negnevitsky M, Muttaqi KM. Short term wind power forecasting using adaptive neuro-fuzzy inference systems. Aust Univ Power Eng Conf (AUPEC) 2007:1–6.
[2] Ortega-Vazquez MA, Kirschen DS. Estimating the spinning reserve requirements in systems with significant wind power generation penetration. IEEE Trans Power Syst 2009;24(1):114–24.
[3] Amjady N, Keynia F, Zareipour HR. A new hybrid iterative method for short-term wind speed forecasting. Eur Trans Electr Power 2011;21:581–95.
[4] Fan S, Liao JR, Yokoyama R, Chen L, Lee WJ. Forecasting the wind generation using a two-stage network based on meteorological information. IEEE Trans Energy Convers 2009;24(2):474–82.
[5] Bessa RJ, Miranda V, Gama J. Entropy and correntropy against minimum square error in offline and online three-day ahead wind power forecasting. IEEE Trans Power Syst 2009;24(4):1657–66.
[6] Kabouris J, Kanellos FD. Impacts of large-scale wind penetration on designing and operation of electric power systems. IEEE Trans Sustainable Energy 2010;1(2):107–14.
[7] Ernst B, Oakleaf B, Ahlstrom ML, Lange M, Moehrlen C, Lange B, et al. Predicting the wind. IEEE Power Energy Mag 2007;5(6):78–89.
[8] Amjady N, Keynia F, Zareipour HR. Wind power prediction by a new forecast engine composed of modified hybrid neural network and enhanced particle swarm optimization. IEEE Trans Sustainable Energy 2011;2(3):265–76.
[9] Atashpaz-Gargari E, Lucas C. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. IEEE Cong Evol Comp (CEC) 2007:4661–7.
[10] Goldberg DE. Genetic algorithms in search optimization and machine learning. Boston: Addison Wesley; 1989.
[11] Sedighizadeh D, Masehian E. Particle swarm optimization methods, taxonomy and applications. Int J Comp Theor Eng 2009;1(5):1793–8201.
[12] Eberhart RC, Kennedy J. Particle swarm optimization, in: Proceeding of the IEEE International Conference on Neural Networks. Perth Aust IEEE Serv Cent 1995, pp. 12–13.
[13] Sideratos G, Hatziargyriou N. Using radial basis neural networks to estimate wind power production. IEEE Power Eng Soc Gen Meet 2007, 10.1109/PES.2007.385812.
[14] Costa A, Crespo A, Navarro J, Lizcano G, Madsen H, Feitosa E. A review on the young history of the wind power short-term prediction. Renew Sustainable Energy Rev 2008;12(6):1725–44.
[15] Lei M, Shiyan L, Chuanwen J, Hongling L, Yan Z. A review on the forecasting of wind speed and enerated power. Renew Sustainable Energy Rev 2009;13(4):915–20.
[16] Ramirez-Rosado IJ, Fernandez-Jimenez LA, Monteiro C, Sousa J, Bessa R. Comparison of two new short-term wind-power forecasting systems. Renew Energy 2009;34(7):1848–54.
[17] Ma L, Luan SY, Jiang CW, Liu HL, Zhang Y. A review on the forecasting of wind speed and generated power. Renew Sustainable Energy Rev 2009;13(4):915–20.
[18] Amjady N, Keynia F, Zareipour H. Wind power prediction by a new forecast engine composed of modified hybrid neural network and enhanced particle swarm optimization. IEEE Trans Sustainable Energy 2011;2(3):265–76.
[19] Liu Y, Shi J, Yang Y, Lee WJ. Short-term wind-power prediction based on wavelet transform-support vector machine and statistic-characteristics analysis. IEEE Trans Ind App 2012;48(2):1136–41.
[20] Khalid M, Savkin AV. A method for short-term wind power prediction with multiple observation points. IEEE Trans Power Syst 2012;27(2):579–86.
[21] An X, Jiang D, Liu C, Zhao M. Wind farm power prediction based on wavelet decomposition and chaotic time series. Exp Syst Appl 2011:11280–538(9) 2011:11280–5.
[22] Kusiak A, Li W. Short-term prediction of wind power with a clustering approach. Renew Energy 2010;35(10):2362–9.
[23] Yan G, Liu Y, Mu G, Cui Y, Li J, Liu J, et al. The ultra-short term prediction of wind power based on chaotic time series. Energy Proc 2012:1490–617(B) 2012:1490–6.
[24] Han S, Li J, Liu Y. Tabu search algorithm optimized ANN model for wind power prediction with NWP. Energy Proc 2011;12:733–40.

[25] Pousinho HMI, Mendes VMF. Catalão JPS. A hybrid PSO–ANFIS approach for short-term wind power prediction in Portugal. Energy Conver and Manage 2011;52(1):397–402.
[26] Zhu B, Chen MY, Wade N, Ran L. A prediction model for wind farm power generation based on fuzzy modeling. Proc Env Sci 2012;12(A):122–9.
[27] Peng H, Liu F, Yang X. A hybrid strategy of short term wind power prediction. Renew Energy 2013;50:590–5.
[28] Foley AM, Leahy PG, Marvuglia A, McKeogh EJ. Current methods and advances in forecasting of wind power generation. Renew Energy 2012;37(1):1–8.
[29] Colak I, Sagiroglu S, Yesilbudak M. Data mining and wind power prediction: a literature review. Renew Energy 2012;46:241–7.
[30] Giorgi MG, Ficarella A, Tarantino M. Error analysis of short term wind power prediction models. Appl Energy 2011;88(4):1298–311.
[31] Bigdeli N, Haeri M, Choobkar S, Jannesari F. Characterization of complex behaviors of TCP/RED computer networks based on nonlinear time series analysis methods. Physica D: Nonlinear Phenom 2007;233(2):138–50.
[32] Bigdeli N, Haeri M. Time-series analysis of TCP/RED computer networks, an empirical study. Chaos Solitons Fractals 2009;39(2):784–800.
[33] Timmer J, Hässier S, Lauk M, Lücking CH. Pathological tremors: deterministic chaos or nonlinear stochastic oscillators. Chaos 2000;10(1):278–88.
[34] Bigdeli N, Sadegh Lafmejani H, Afshar K. Characterization and short-term predictability analysis of wind speed and wind power time series. J Wind Eng Indust Aerodyn 2013; Submitted for publication.
[35] Eckmann J-P, Kamphorst SO, Ruelle D, Ciliberto S. Liapunov exponents from time series. Phys Rev A 1986;34:4971–9.
[36] Schuster HG. Handbook of chaos control. Singapore: Wiley VCH; 1999.
[37] Casrellini H, Romanelli L. Application of recurrence quantified analysis to study the dynamics of chaotic chemical reaction. Physica A 2004;342:301–7.
[38] Andrade KC, Wehrle R, Spoormaker VI, Samann PG, Czisch M. Statistical evaluation of recurrence quantification analysis applied on single trial evoked potential studies. Clinic Neurophy 2012;123(8):1523–35.
[39] Aschero G, Gizdulich P. Denoising of surface EMG with a modified Wiener filtering approach. J Electromyog Kines 2010;20(2):366–73.
[40] Ng WW, Panu US, Lennox WC. Chaos based analytical techniques for daily extreme hydrological observations. J Hydro 2007;342:17–41.
[41] Armienti P, Gasperini D. Isotopic evidence for chaotic imprint in upper mantle heterogeneity. Geochem Geophys Geosyst 2010;11(5):Q0AC02.
[42] Bigdeli N, Jafarzadeh M, Afshar K. Characterization of Iran stock market indices using recurrence plots. Proc Int Conf Manage Serv Sci (MASS) 2011:1–5.
[43] Bigdeli N, Afshar K, Amjady N. Market data analysis and short-term price forecasting in the Iran electricity market with pay-as-bid payment mechanism. Electr Power Syst Res 2009;79:888–98.
[44] Bigdeli N, Afshar K. Chaotic behavior of price in the power markets with pay-as-bid payment mechanism. Chaos Solitions Fractals 2009;42(4):2560–9.
[45] Bigdeli N, Afshar K. Characterization of Iran electricity market indices with pay-as-bid payment mechanism. Physica A 2009;388(8):1577–92.
[46] Hegger R, Kantz H, Schreiber T. Practical implementation of nonlinear time series methods: the TISEAN package. Chaos 1999;9:413–38.
[47] Marwan MC, Romano M, Thiel M, Kurths J. Recurrence plots for the analysis of complex systems. Phys Rep 2007;438:237–329.
[48] Romano MC, Thiel M, Kurths J, Bloh W. Multivariate recurrence plots. Phys Lett A 2004;330:214–23.
[49] CRP toolbox for Matlab provided by TOCSY. Available from: 〈http://tocsy.agnld.uni-potsdam.de〉, March 2013.
[50] Catalão JPS, SJPS Mariano, Mendesb VMF. Ferreira LAFM. Short-term electricity prices forecasting in a competitive market: a neural network approach. Electr Power Syst Res 2007;77:1297–304.
[51] Haykin S. Neural Networks: A Comprehensive Foundation. New Jersey: Prentice-Hall; 1999.
[52] Almeida LB. In: Fiesler E, Beale R, editors. Multilayer perceptrons, handbook of neural computation. New York: Oxford University Press; 1997.
[53] Szkuta BR, Sanabria LA, Dillon TS. Electricity price short-term forecasting using artificial neural networks. IEEE Trans Power Syst 1999;14(3):851–7.
[54] Abdechiri M, Faez K, Bahrami H. Neural network learning based on chaotic imperialist competitive algorithm. 2nd Int Work Intel Syst Appl (ISA) 2010:1–5.
[55] Niknam T, Taherian Fard E, Pourjafarian N, Rousta A. An efficient hybrid algorithm based on modified imperialist competitive algorithm and K-means for data clustering. Eng Appl Art Intel 2011;24(2):306–17.
[56] Jolai F, Sangari M, Babaie M. Pareto simulated annealing and colonial competitive algorithm to solve an offline scheduling problem with rejection. J Eng Manuf 2010;224(7):1119–31.
[57] Lucas C, Nasiri-Gheidari Z, Tootoonchian F. Application of an imperialist competitive algorithm to the design of a linear induction motor. Energy Conver Manage 2010;51(7):1407–11.
[58] Tayefeh Mahmoudi M, Forouzideh N, Lucas C, Taghiyareh F. Artificial neural network weights optimization based on imperialist competitive algorithm. 7th Int Conf Comp Sci Inf Tech 2009.
[59] Shahidehpour M, Yamin H, Li Z. Market operations in electric power systems: forecasting, scheduling, and risk management. New York: John Wiley & Sons Inc; 2002.
[60] The MathWorks, MATLAB. Available from: 〈http://www.mathworks.com〉, March 2013.